Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.201961465
Vol. 6, No. 6, Desember 2019, hlm. 697-702 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 30/E/KPT/2018 e-ISSN: 2528-6579

IMPLEMENTASI TEKNIK ENKRIPSI DAN DEKRIPSI DI FILE VIDEO MENGGUNAKAN ALGORITMA BLOWFISH

Nuniek Fahriani[1], Harunur Rosyid[2]

[1,2]Universitas Muhammadiyah Gresik
Email: [1]nuniekfahriani@umg.ac.id, [2]harun@umg.ac.id

(Naskah masuk: 18 Desember 2018, diterima untuk diterbitkan: 25 April 2019)

Abstrak

Kriptografi (*cryptography*) merupakan proses keamanan data untuk menjaga pesan (file) agar tidak "diganggu" oleh pihak ketiga. kriptografi memiliki unsur proses, yaitu : enkripsi, dekripsi, dan kunci. Menjadi kebutuhan *user* untuk menghindari 'pihak ketiga' yang bisa merubah, mengambil ataupun menghilangkan data secara fisik atau menjalankan fungsi program yang mengganggu sistem. Tingkat keaslian data menjadi bagian penting didalam sistem keamanan data. Jenis data berupa file yang berpotensi "dirusak" secara illegal tidak hanya berextension .doc bisa saja menyerang jenis file yang berextension selain .doc. Dalam penelitian ini, proses keamanan yang dilakukan pada file berextension file video. Untuk menjalankan fungsi dari sistem keamanan data file video, legalitas akses akan data sangat penting untuk *secure* sehingga tidak berakibat kepada penyalahgunaan hak akses data. Teknik yang digunakan untuk menunjang enkrip dan dekrip file video adalah menerapkan algoritma blowfish didalam implementasinya. Algoritma ini memiliki sistem keamanan yang variabel. Hasil ujicoba menggunakan 6 contoh file extension yang melalui teknik enkrip dan dekrip adalah : file extension .asf, .wmv, .avi, .3pp, .flv, .vob.

Kata kunci: *Kriptografi, enkripsi, dekripsi, blowfish, J2SE*

IMPLEMENTATION ENCRYPTION AND DECRYPTION OF VIDEO FILE USING BLOWFISH ALGORITHM

Abstract

Cryptography (cryptography) is a data security process to keep messages (files) from being "disturbed" by third parties. cryptography has a process element, namely: encryption, decryption, and key. It becomes the user's need to avoid 'third parties' which can change, retrieve or eliminate physical data or run program functions that interfere with the system. The level of authenticity of the data is an important part of the data security system. The type of data in the form of files that have the potential to be "corrupted" illegally does not only have an extension. Doc can attack file types with extensions other than .doc. In this study, the security process carried out on file extension video files. To perform the functions of the video file data security system, the legality of access to data is very important to secure so that it does not result in misuse of data access rights. The technique used to support the encryption and decryption of video files is to apply the blowfish algorithm in its implementation. This algorithm has a variable security system. The test results using 6 sample file extensions through the encryption and decryption techniques are: extension file .asf, .wmv, .avi, .3pp, .flv, .vob.

Keywords: *Kriptography, encryption, decryption, blowfish, J2SE*

1. PENDAHULUAN

Security network (kemanan komputer), menjadi bagian penting didalam sistem, baik yang berupa data (dokumen, video, audio, gambar), fisik ataupun *logic* (bahasa pemrograman). Saat ini, jenis serangan tidak hanya melalui media *online* (berbasis internet), yang berbasis aplikasi *desktop* pun bisa juga di ganggu dari *user* yang tidak bertanggungjawab. Pengertian dari aplikasi desktop adalah suatu aplikasi yang sistemnya *standalone* (sendiri) tanpa menggunakan browser atau koneksi internet disuatu komputer otonom. Banyak celah yang bisa ditembus antara lain : faktor kelemahan manusia, kelemahan sistem, dan kelemahan aplikasi. Keamanan sebuah informasi merupakan suatu hal yang harus diperhatikan. Masalah tersebut penting karena jika sebuah informasi dapat di akses oleh orang yang tidak berhak atau tidak bertanggung jawab, maka

keakuratan informasi tersebut akan diragukan, bahkan akan menjadi sebuah informasi yang menyesatkan. (Paryati, 2008)

Menjadi kebutuhan *user* untuk menghindari 'pihak ketiga' yang bisa merubah, mengambil ataupun menghilangkan data secara fisik atau menjalankan fungsi program yang mengganggu sistem. Terdapat berbagai bentuk pesan rahasia seperti pesan teks (dalam bentuk file), pesan citra, pesan audio dan pesan video yang umum digunakan. Banyak cara atau metode yang digunakan untuk mengamankan informasi atau data agar tidak jatuh ke tangan pihak-pihak yang tidak berkepentingan. Sebagai contoh adalah teknik kriptografi. (Indriyono, 2016) Tingkat keaslian data menjadi bagian penting didalam sistem keamanan data. Jenis data berupa file yang berpotensi "dirusak" secara illegal tidak hanya berextension .doc bisa saja jenis file yang berextension file video, mengapa file video juga perlu adanya sistem keamanan? Karena dalam file video banyak sekali unsur yang terlibat visualisasinya. Jika ada yang merubah, merusak, atau bahkan mengambil file secara illegal maka visualisasi dari file video akan menjadi rusak dan tidak akan utuh lagi. Sehingga berakibat menimbulkan efek negatif bagi *user* yang memiliki *authorize* legalitas file video tersebut.

Dalam menjalankan fungsi dari sistem keamanan data file video, legalitas akses akan data sangat penting untuk *secure* sehingga tidak berakibat kepada penyalahgunaan hak akses data, maka diperlukan teknik yang bisa menjalankan fungsi tersebut yaitu menggunakan proses enkrip dan dekrip di file video. Algoritma yang digunakan menggunakan algoritma blowfish. Algoritma ini memiliki sistem keamanan yang variabel.

Keseluruhan proses dalam membuat aplikasi keamanan data berbasis *desktop* melalui proses enkrip dan dekrip menggunakan algoritma blowfish, mekanismenya mengubah teks asli (*plaintext*) file video kemudian dirubah kedalam bentuk *ciphertex* (tidak dapat dibaca / dibuka) yaitu di-enkrip yang selanjutnya mengembalikan kembali dalam file asal dengan cara men-dekrip. Sesuai dengan kebutuhan *sender* (pengirim pesan) ke *receiver* (penerima pesan) dan adanya legalitas *authorize* yang berupa kunci (*key*) yang diberikan oleh *sender* ke *receiver*.
Dibawah ini gambaran dari beberapa ancaman didalam sistem keamanan data, seperti yang terlihat pada gambar 1.

2. KRIPTOGRAFI

Kriptografi (*cryptography*) merupakan proses keamanan data untuk menjaga pesan (file) agar tidak "diganggu" oleh pihak ketiga. (*Cryptography is the art and science of keeping messages secure*) (Stallings, 1995). "*Crypto*" berarti "*secret*" (rahasia) dan "*graphy*" berarti "*writing*" (tulisan).

Awal mula kriptografi dipahami sebagai ilmu tentang menyembunyikan pesan (Sadikin, 2012). Para pengguna dan user pengguna kriptografi disebut *cryptographers*. Sebuah algoritma kriptografik (*cryptographic algorithm*), disebut *cipher*, adalah sebuah persamaan matematika pada proses teknik enkrip dengan dekrip. Untuk implementasi proses teknik persamaan matematik (untuk enkrip dan dekrip) tersebut memiliki keterkaitan proses matematisnya.

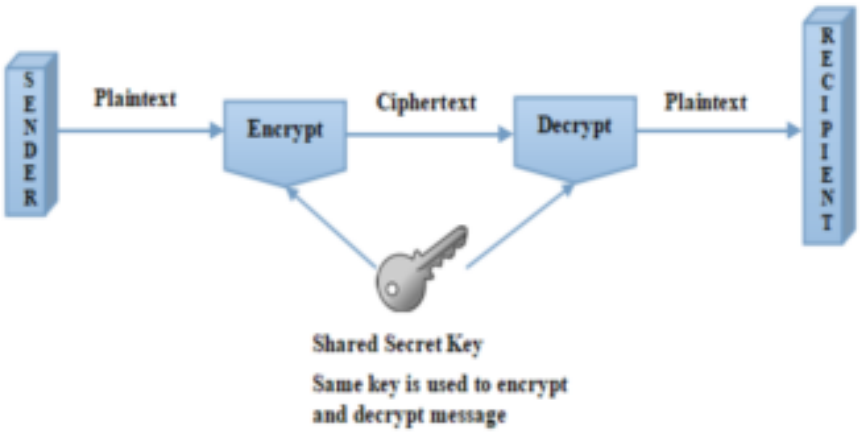
Gambar 1. Jenis Serangan

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah enkrip (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, istilah yang biasa digunakan yaitu "*encipher*". Proses kebalikan, untuk mengubah *ciphertext* menjadi *plaintext*, disebut dekrip (*decryption*). Menurut ISO 7498-2, istilah yang biasa digunakan untuk proses ini adalah "*decipher*".

Menurut (Ariyus, 2008), kriptografi memiliki tiga fungsi dasar. Yaitu : enkripsi, dekripsi, dan kunci. Definisi lain dari kriptografi adalah ilmu tentang penggunaan matematika (Kaur, 2012) untuk membuat informasi teks biasa (P) menjadi format teks *cipher* (C) yang tidak terbaca yang disebut enkripsi dan mengubah kembali teks sandi tersebut menjadi teks biasa yang disebut dekripsi dengan himpunan aturan menyandikan pesan (E) dengan kunci enkrip (k1 dan k2) dan algoritma dekrip (D) yang membalik dan menghasilkan teks asli dari teks sandi. Hal ini dapat diartikan sebagai teks *Cipher* C = E {P, Key} dan teks polos C = D {C, Key}(Bhardwaj et al., 2016). Dibawah ini digambarkan proses enkripsi dan dekripsi data.

3. ALGORITMA BLOWFISH

Proses penerapannya menggunakan algoritma blowfish sebagai fungsi kriptografinya, dimana algoritma ini dibuat oleh seorang *Cryptanalyst* bernama Bruce Scheneir, Presiden perusahaan *Counterpane Internet Security*, Inc (Perusahaan konsultan tentang kriptografi dan keamanan komputer) dan dipublikasikan tahun 1994.



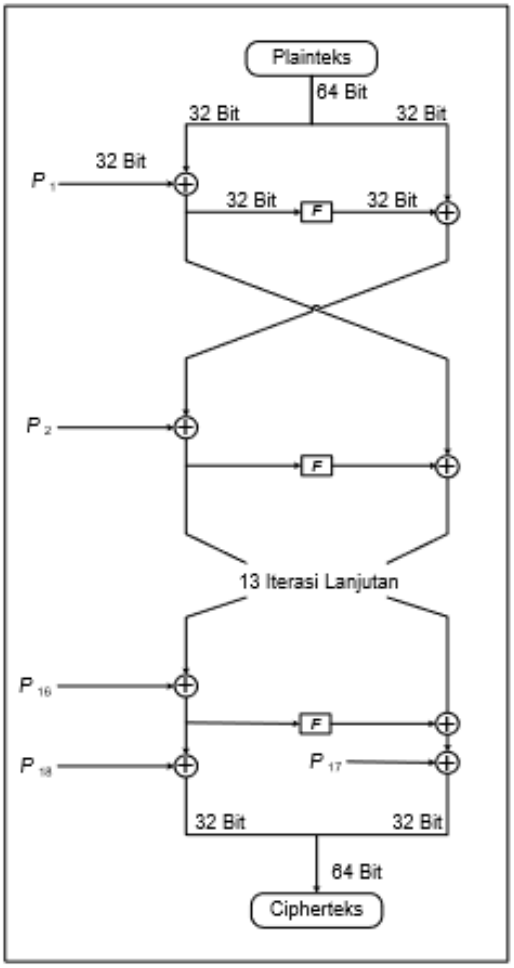
Gambar 2. Proses Enkripsi dan Dekripsi

Algo Blowfish adalah metode enkripsi yang mirip dengan DES yang dirancang untuk mikroprosesor besar (32 bit ke atas dengan *cache* data yang besar). Blowfish dikembangkan untuk memenuhi kriteria desain berikut :

1. Kecepatan, dalam proses yang optimal algo blowfish dapat mencapai kecepatan sampai 26 *clock cycle* per *byte*.
2. Proses penyimpanan (kecepatan sistem), algo blowfish dapat melakukan proses di memori setidaknya kurang dari 5KB.
3. Sederhana, blowfish hanya menggunakan operasi yang mudah dipahami : penambahan (*addition*), XOR, dan penelusuran tabel (*table lookup*) pada operand 32 bit.
4. Perancangan desain sistem mudah untuk dianalisa yang membuatnya terdeteksi terhadap kesalahan implementasi.
5. Variasi keamanan, karakteristik key blowfish dapat bervariasi dan dapat mencapai kurang lebih sampai 448 bit (56 byte). *End Of File* (EOF) *End Of File* (EOF) ialah bagian dari teknik yang digunakan pada keamanan steganografi. Proses bagian tersebut digunakan dengan metode menyisipkan data pada akhir file. Blowfish dirancang mengacu pada teknik jaringan feistel yang terdiri dari 16 putaran. Inputan dengan nilai 64 bit, X. Bentuk pola jaringan feistel digambarkan dibawah ini.

Alur algo enkripsinya adalah :

- Array P terdiri dari 18 kunci 32-bit sub kunci : P1,P2,…….,P18
- Empat 32-bit S-box masing-masing mempunyai 256 entri :
  S1,0,S1,1,……..,S1,255
  S2,0,S2,1,……..,S2,255
  S3,0,S3,1,……..,S3,255
  S4,0,S4,1,……..,S4,255
- *Plaintext* yang akan dienkripsi diasumsikan sebagai masukan, *Plaintext* tersebut diambil sebanyak 64-bit, dan apabila kurang dari 64-bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
- Hasil nya dibagi 2, 32-bit pertama disebut XL, 32-bit yang kedua disebut XR.

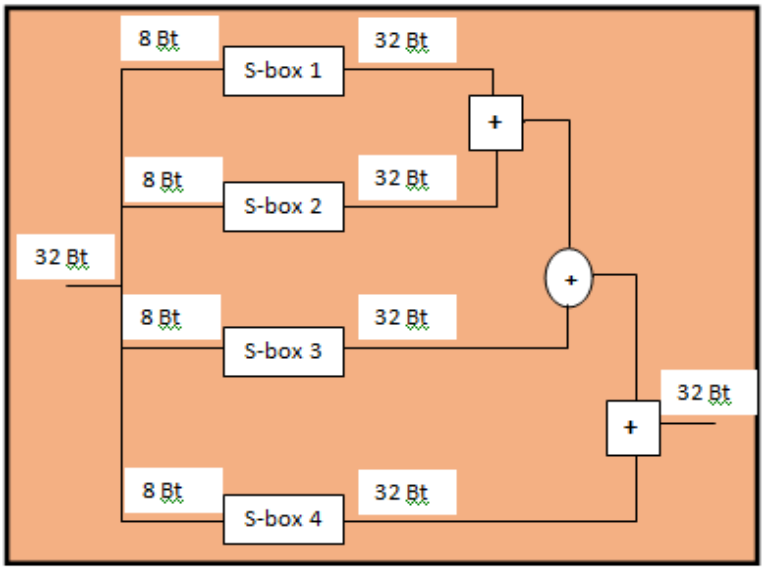


Gambar 3. Pola Jaringan Feistel

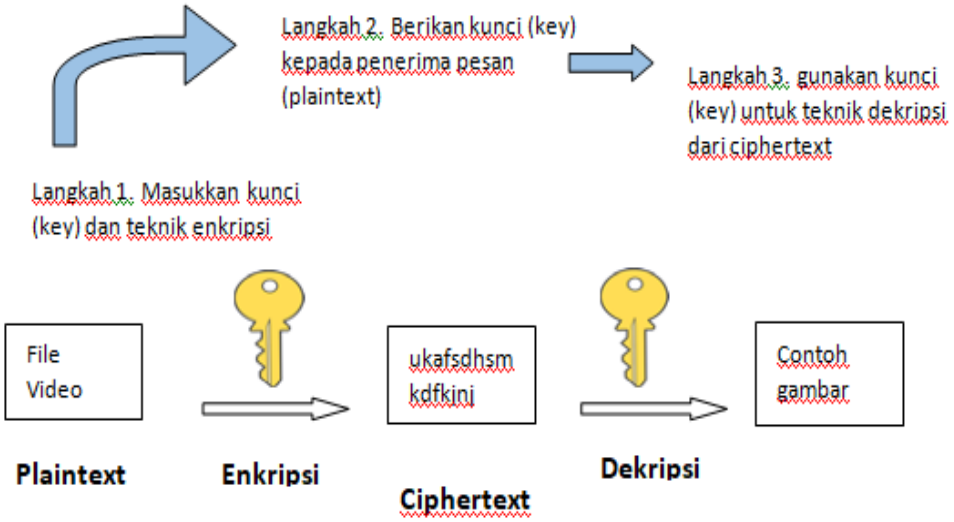
- Selanjutnya lakukan operasi :
  XL = XL xor Pi dan
  XR = F(XL) xor XR

- Hasil dari operrasi diatas ditukar XL menjadi XR dan XR menjadi XL.
- Lakukan perulangan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran XL dan XR.
- Proses ke-17 lakukan operasi untuk : XR = XR xor P17 dan XL = XL xor P18.
- Terakhir satukan kembali XL dan XR sehingga menjadi 64-bit (kembali).

Pada teknik Dekrip, langkahnya sama dengan teknik enkrip sebelumnya, hanya urutan Pbox digunakan berdasarkan urutan kebalikan. Langkah diatas pada bentuk jaringan feistel telah dituliskan mengenai penggunaan fungsi F. Dimana bagi X1 menjadi empat bagian 8-bit : a,b,c,d. F(X1) = ((S1,a+S2,b mod 232) XOR S3,c)+S4,d mod 232. Untuk lebih jelas tentang Fungsi F dapat dilihat pada gambar dibawah ini, yaitu :



Gambar 4. Fungsi F



Gambar 5. Ilustrasi Langkah Enkripsi Dekripsi

4. IMPLEMENTASI ENKRIPSI DESKRIPSI DI FILE VIDEO

Metode implementasi untuk teknik enkrip dan dekrip pada aplikasi ini yaitu algoritma Blowfish. Aplikasi ini berfungsi apabila kita ingin mengenkripsi *file* video yang bersifat rahasia, ilustrasi langkah penerapan teknik enkripsi dan dekripsi terlihat pada gambar 5.

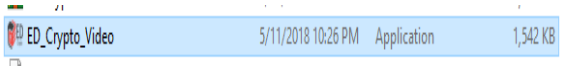
Kebutuhan perangkat untuk implementasi, Hardware : Laptop HP dengan spesifikasi : Procesor Intel Core i3-6006U, RAM 4.00 GB, 64-bit OS. Software: Ms. Office, Ms. Excel, Windows 10, J2SE Netbean IDE 8.0.2. dan dibawah ini extension file video yang digunakan untuk ujicoba implementasi yaitu pada tabel.1 dibawah ini :

Tabel 1. File Video

No	Nama File
1.	.asf
2.	.avi
3.	.wmv
4.	.3pp
5.	.flv
6.	.vob

Langkah-langkah implementasi aplikasi dapat digambarkan sebagai berikut :

1. Langkah pertama *Double click* utk file *exe.*ED_Crypto_Video, gambar *exe* seperti yang terlihat dibawah ini :

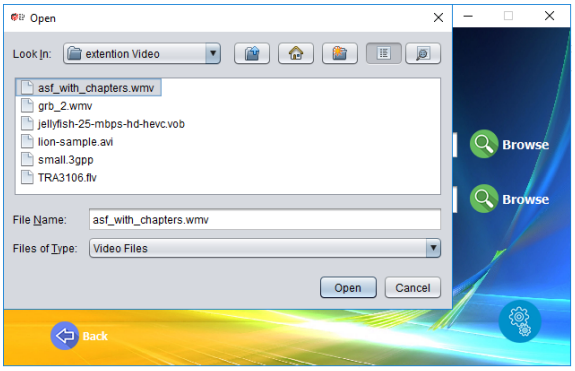


Gambar 6. File Exe

2. Selanjutnya yaitu tampilan awal dari teknik enkripsi dan dekripsi file video, yaitu :

Gambar 7. Halaman Depan

Untuk proses enkripsi, akan kita pilih file video yang akan dienkripsi, gambarnya adalah :



Gambar 8. Menu Pilihan File Video

Maka akan keluar permintaan untuk pemberian kunci (*key*). Kunci yang dimasukkan harus sama dengan kunci untuk saat proses enkripsi dan dekripsi. *user* yang menerima *(receiver)* file yang telah terenkripsi, selanjutnya akan diberikan kunci yang sama oleh *sender* untuk dapat membuka file video yang terenkripsi. Berikut dibawah ini adalah gambar proses menu enkripsi, yaitu :

Gambar 9. Proses Enkripsi

Gambar 10. Proses Dekripsi

Hasil file yang terenkripsi yang tidak dapat dibaca (bentuk *ciphertext*) pada folder dimana file video disimpan/diletakkan terlihat pada gambar 11. Dan gambar 12 merupakan proses dekripsi yaitu kembalinya file asli (*plaintext*) yang dapat dibaca oleh penerima pesan. Yaitu :

Gambar 11. Hasil Enkripsi Ciphertext

Gambar 12. Hasil Dekripsi Plaintext

Dibawah ini, kami sajikan data hasil dari implementasi extension file video dengan mengambil beberapa contoh (6 contoh) dari file yang berextension file video pada Tabel 2.

Tabel 2. Hasil Ujicoba 6 File berextension file video

No	Plaintext (Nama File Video)	Ukuran File Enkripsi (Byte)	Ukuran File Dekripsi (Byte)	Ciphertext
1.	Asf_with_chapters.asf	16.383.000	16.382.993	15.62 MB
2.	Grb_2.wmv	880.888	880.887	860.24 KB
3.	Lion_sample.avi	3.71	3.71	3.72 MB
4.	Small.3pp	344.176	344.169	336.11 KB
5.	TRA3106.flv	3.225.896	3.225.894	3.08 KB
6.	Jellyfish-25-mbps-hd-hevc.vob	14.145.544	14.145.536	13.49 MB

Didalam mekanisme keamanan data kita harus memegang prinsip-prinsip dibawah ini :

A. Bersifat rahasia (*confidentiality*), dimana file dapat tersimpan dengan aman tanpa ada ancaman dari pihak yang tidak berhak mengakses data tersebut.

B. Keaslian terjaga (*Integrity*), bahwa file tetap berupa file aslinya yaitu terjaga kerahasiaanya, tidak dirubah atau dimodifikasi dalam perjalanannya dari *source* (sumber) ke *destination* (tujuan).

C. Ketersediaan (*Availability*), yaitu *brainware* adalah orang (*user*) yang memiliki hak akses atau *authorized users*, dimana diberi akses sesuai kunci dari pengguna dan tidak ada manipulasi.

5. KESIMPULAN

1. Proses yang dilakukan merupakan implementasi dari teknik enkripsi dan dekripsi untuk pengamanan data file video menggunakan Algoritma Blowfish.
2. Ukuran file dan waktu proses akan muncul sebagai informasi tambahan bahwa besar file yang telah dienkripsi berapa MB/KB dalam waktu berapa detik.
3. *user* penerima pesan (file) yaitu *receiver* harus juga mempunyai aplikasi *exe.*ED_Crypto_Video, karena aplikasi ini sifatnya *standalone* berbasis *desktop*.

Berdasarkan keseluruhan proses dalam membuat aplikasi keamanan data berbasis desktop dengan implementasi enkripsi dan dekripsi menggunakan algoritma blowfish, bahwa aplikasi ini telah berhasil mengubah teks asli fie video kemudian dirubah dalam bentuk *ciphertex* (tidak dapat dibaca / dibuka) yaitu meng-enkrip yang selanjutnya mengembalikan kembali dalam file asal dengan cara men-dekrip. Saran untuk ujicoba, bisa ditambahkan data berbagai jenis file extension lainnya. Sehingga bisa disajikan perbedaan besaran file dalam proses enkripsi dan dekripsi.

DAFTAR PUSTAKA

ARIYUS, D., 2008. *Pengantar ilmu kriptografi: teori analisis & implementasi*. Penerbit Andi.

BHARDWAJ, A., SUBRAHMANYAMB, G., AVASTHIC, V. & SASTRYD, H., 2016. Security Algorithm For Cloud Computing. In *International conference on computational modeling and security (CMS)*., 2016. Elsevier.

INDRIYONO, B.V., 2016. Implementasi Sistem Keamanan File dengan Metode Steganografi EOF dan Enkripsi Caesar Cipher. *Jurnal Sisfo*, 06(01), pp.1–16.

KAUR, S., 2012. Cryptography and encryption in cloud computing. *VSRD International journal of Computer science and Information Technology*, Vol. 2(3), pp.242-49.

PARYATI, 2008. Keamanan Sistem Informasi. In *Seminar Nasional Informatika 2008 (semnasIF 2008)*. Yogyakarta, 2008. UPN "Veteran" Yogyakarta.

SADIKIN, R., 2012. *Kriptografi Untuk keamanan Jaringan*. Penerbit ANDI.

STALLINGS, W., 1995. *Network and Internetwork Security*. Prentice Hall.